

**REMARKS**

Please reconsider the application in view of the following remarks. Applicant thanks the Examiner for courtesies extended during the Examiner Interview on September 6, 2007, and for carefully considering this application.

**Disposition of Claims**

Claims 1, 3, 5, 8, 21, 22, 24, and 26-32 are pending in the referenced application.

**Rejection Under 35 U.S.C. § 103**

Claims 1, 3, 5, 8, 21, 22, 24, and 26-32 stand rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent Application Publication No. 2002/0095660 ("O'Brien") in view of U.S. Patent No. 6,678,883 ("Berry") and U.S. Patent No. 5,655,121 ("Delagi"). The rejection is respectfully traversed for at least the following reasons.

**I. NONE OF THE CITED PRIOR ART TEACHES OR SUGGEST AN OBJECT FILE AS RECITED IN THE CLAIMS**

Independent claims 1 and 21 recite, in part,

wherein the object file comprises a linear sequence of bytes comprising:  
a file header,  
a first section header of type enable control block (ECB),  
a second section header of type action,  
a first section data entry associated with the first section header comprising a name of a probe to enable and a name of a second section data entry of type action, and

the second section data entry associated with the second section header defining at least one selected from a group consisting of a predefined action and a name of a third section data entry of type program object code corresponding to an action

The above limitations define the structure of the object file. As claimed, **the object file is used to enable probes and defines actions to perform when an enabled probe is encountered during tracing.** With respect to the object file, the object file is a linear sequence of bytes (*see e.g.*, instant specification, [0020] and Figure 2). The linear sequence of bytes includes: (i) a file header (*see e.g.*, instant specification, [0021]); (ii) a first section file header of type ECB (*see e.g.*, instant specification, [0030]); (iii) a second section header of type action (*see e.g.*, instant specification, [0026] and Figure 3); and (iv) a first section data entry associated with the first section header and a second section data entry associated with the second section header (*see e.g.*, instant specification, [0025] and Figure 4).

With respect to the section data entries, the first section data entry defines the name of probe to enable and the name of the second section data entry of type action. Said another way, the first section data entry **links** the probe with the action to perform when the probe is encountered (*see e.g.*, claim 1: "...executing at least one selected from a group consisting of the predefined action and the action associated with the probe to obtain data"). Further, the second section data entry includes either the predefined action (*i.e.*, the action to perform when the probe is encountered) or a name of a third section data entry that includes the program object code corresponding to the action (*i.e.*, the program object code to be executed when the probe is encountered).

Turning to the cited prior art, the Examiner admits that O'Brien fails to teach or suggest the object file recited in the claims. See Office Action mailed July 9, 2007, p. 3. Instead the Examiner relies on Berry to teach that which O'Brien lacks. As discussed during the Examiner Interview, Berry only discloses the general concept of tracing but fails to disclose the structure of the object file as recited in the claims.

Specifically, Berry discloses using a trace program to record data upon the execution of a hook. The recorded data is subsequently stored in a buffer or a trace file. See Berry, col. 10, ll. 30-47.<sup>1</sup> Further, the invention disclosed in Berry "employs trace hooks that aid in the identification of modules that are used in an application under trace. With Java operating systems, the present invention employs trace hooks that aid in identifying loaded classes and methods." Berry, col. 10, ll. 48-52. Berry then discloses the use of a symbol file that "contains information useful in performing symbolic resolution of address information in trace files for each instance of a module." See Berry, col. 3, ll. 55-59. "During post processing of the trace information generated by a performance trace of a computer program, symbolic information stored in the merged symbol file is compared to the trace information stored in the trace file. The post processing typically occurs shortly after the trace or at some remote time after the trace of the computer program." See Berry, col. 3, ll. 60-65.

In view of the above, Berry does not teach or suggest an object file that is used to enable probes for tracing an application and that includes information about the action to perform

---

<sup>1</sup> "Trace program 400 may be used to record data upon the execution of a hook, which is a specialized piece of code at a specific location in a routine or program in which other routines may be connected. Trace hooks are typically inserted for the purpose of debugging, performance analysis, or enhancing functionality. These trace hooks are employed to send trace data to trace program 400, which stores the trace data in buffer 404. The trace data in buffer 404 may be subsequently stored in a file for post-processing, or the trace data may be processed in real-time. The trace data in either

when a given probe is enabled. Rather, as discussed, Berry focuses on the processing of the information resulting from the tracing (*i.e.*, information obtained from the execution of the trace hooks).

Finally, Degali fails to teach that which O'Brien and Berry lack. Specifically, while Degali is directed to tracing an application, Degali focuses on the insertion of probes into the source code of an application, *see e.g.*, Degali col. 4-5. However, Degali is silent with respect to an object file that includes the structure recited in the claims or the functionality to enable probes and define actions to take when an enabled probe is encountered.

In view of the above, none of the cited prior art teaches or suggest an object file as recited in independent claims 1 and 21.

## II. NONE OF THE CITED PRIOR ART TEACHES OR SUGGESTS A PROBE DESCRIPTION AS RECITED IN THE CLAIMS

Dependent claims 5 and 26 recite, in part, "wherein the trace source code comprises a probe description associated with the probe, wherein the probe description comprises an optional predicate and the action." The Examiner has asserted that Degali teaches or suggest the above limitation. The Applicant respectfully disagrees as Degali is silent with respect to including a probe description into the source code where the probe description includes an optional predicate. Further, neither O'Brien nor Berry teach or suggest a probe description as recited in the claims. In view of the above, none of the cited prior art teaches or suggest a probe description as recited in dependent claims 5 and 26.

---

the buffer 404 or the trace file, is then processed by the post-processor 406 to generate an indexed database of symbolic

III. NONE OF THE CITED PRIOR ART TEACHES OR SUGGESTS A THIRD SECTION HEADER AND A FOURTH DATA ENTRY AS RECITED IN THE CLAIMS

Dependent claims 29 and 32 recite, in part, "a third section header comprises a name of a fourth section data entry comprising program object code associated with a predicate, wherein the program object code associated with the predicate is executed when the probe is encountered during tracing."

As discussed above, the Examiner admits that O'Brien fails to teach or suggest an object file as recited in the claims. Further, as discussed above, Berry also fails to teach or suggest an object file having the structure as recited in independent claims 1 and 21. From this it logically follows that Berry cannot disclose the additional section header or section data entry of the object file as Berry does not teach or suggest section headers or section data entries description as recited in dependent claims 29 and 32. Finally, Degali also fails to teach or suggest that which O' Brien and Berry lack. Specifically, as discussed, Degali only discloses the insertion of probes into the source code of an application but does not teach or suggest an object file having the requisite structure or functionality of the object file recited in independent claims 1 and 21. Based on this, Degali cannot possibly disclose the additional section header or section data entry of the object file dependent claims 29 and 32.

In view of the above, none of the cited prior art teaches or suggest the additional section header or section data entry as recited in dependent claims 29 and 32.

---

data for loaded modules, as described more fully hereafter." Berry, col. 10, ll. 30-47

#### IV. CONCLUSION

In view of the above, the pending claims are patentable over all of the cited prior art.

#### Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/345001).

Dated: September 10, 2007

Respectfully submitted,

By /Robert P. Lord/

Robert P. Lord  
Registration No.: 46,479  
OSHA · LIANG LLP  
1221 McKinney St., Suite 2800  
Houston, Texas 77010  
(713) 228-8600  
(713) 228-8778 (Fax)  
Attorney for Applicant